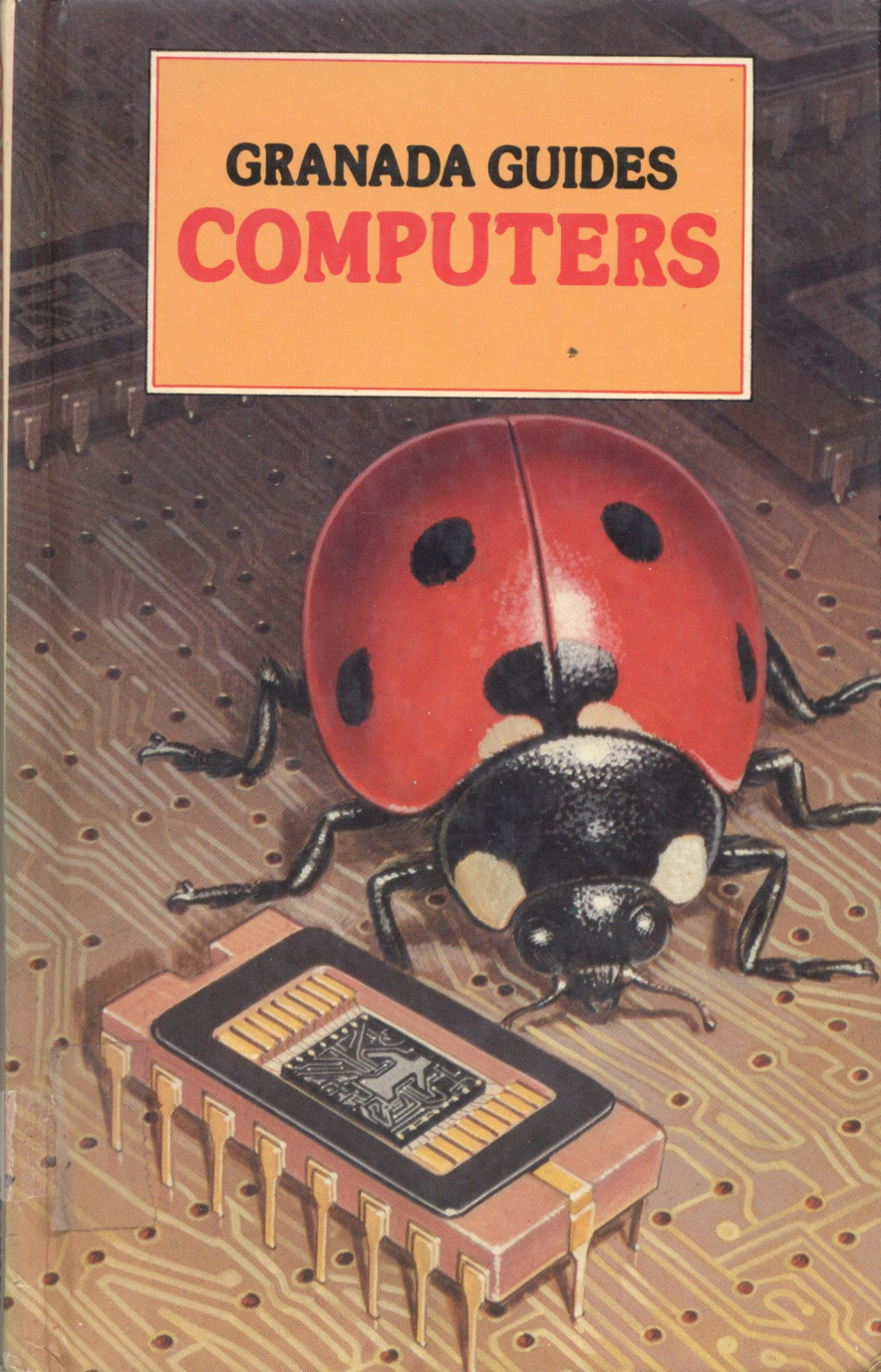
GRANADA GUIDES

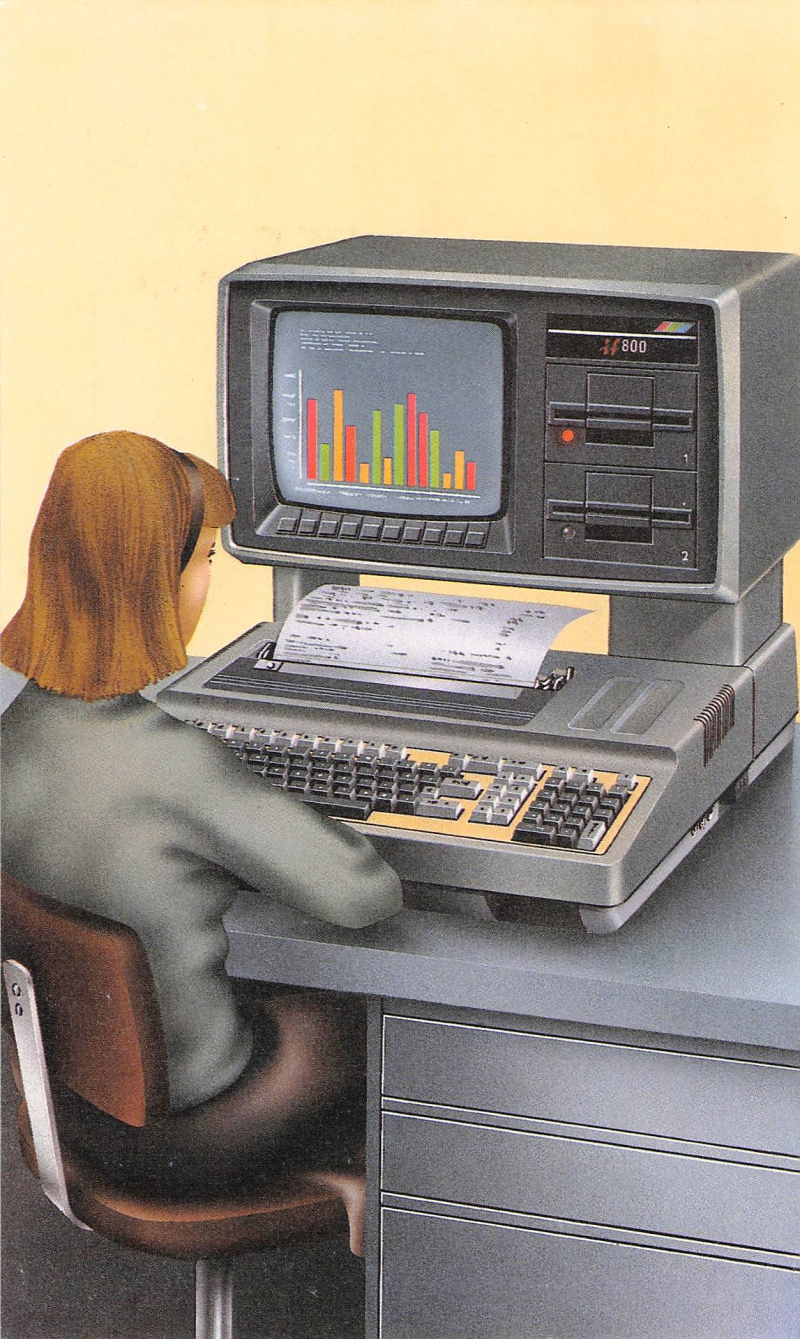
COMPUTERS



WITHDRAWN

7257

COMPUTERS


800

COMPUTERS

by Tom Fry

Illustrated by Janos Marrfy,
Roy Woodard and Mary Tomlin

GRANADA
London Toronto Sydney New York

ISLE OF WIGHT
COUNTY COUNCIL
Class J 519.4 Acc. No. 236362

Granada Publishing Limited
Frogmore, St Albans, Herts AL2 2NF
and
36 Golden Square, London W1R 4AH
866 United Nations Plaza, New York, NY 10017, USA
117 York Street, Sydney, NSW 2000, Australia
100 Skyway Avenue, Rexdale, Ontario, M9W 3A6, Canada
61 Beach Road, Auckland, New Zealand

Published by Granada Publishing 1982

Copyright © Granada

British Library Cataloguing in Publication Data

Fry, Tom
Computers. – (Granada guide series; 23)
1. Electronic digital computers – Juvenile literature
I. Title
001.64 QA76.52

ISBN 0 246 11895 4

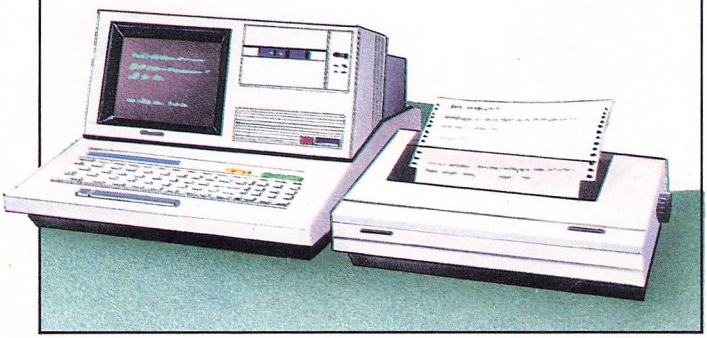
Printed and bound in Great Britain by
William Collins Sons & Co Ltd, Glasgow

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Granada ®
Granada Publishing ®

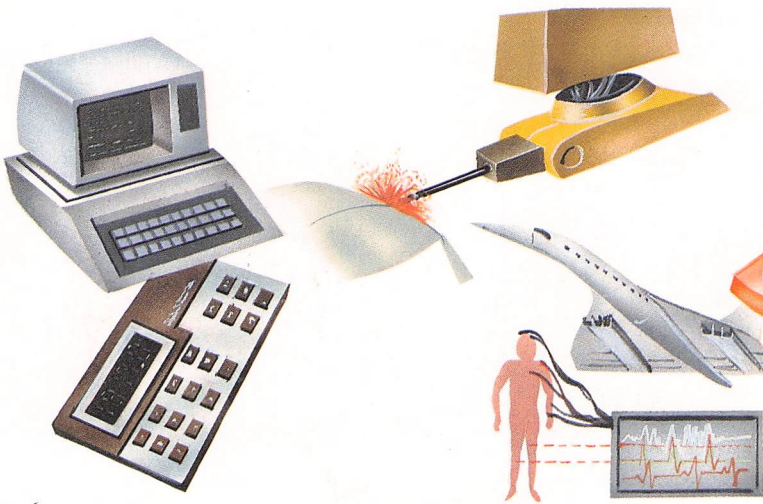
Contents

Computers All Around	6
Computer Development	8
The Elements of a Computer	13
Communicating With a Computer	18
The Central Processor	30
Information Storage	41
Computer Programs	45
Computers in Action	50
The Magic Chip	55
Glossary of Terms	62



Computers All Around

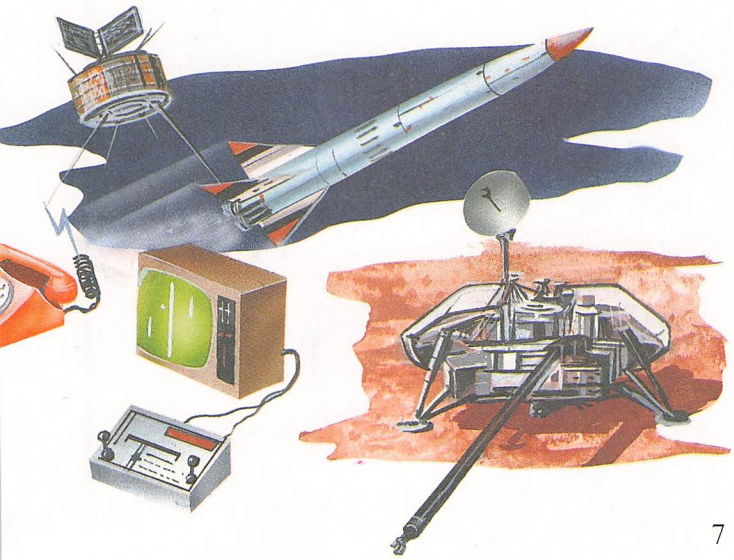
Until about thirty years ago few people had heard about computers and fewer still really knew anything about them. Today, while many people still may not know a lot about them, it would be difficult to find anyone who had not heard of computers. These machines play an ever-increasing role in almost every aspect of our daily lives. For instance, when shopping you can be fairly sure that a computer has been involved at some stage in the production of whatever you buy. And the shop from which you bought the goods could well use a computer to record their sale. This is why more and more packets and tins carry a group of narrow lines on the label. This is a bar code and is there so that a computer can read and record what is sold. In the home, many of



the letters which arrive have been addressed by computers, and most of the domestic bills, such as electricity, gas and telephone charges, have been calculated by computer.

The influence of computers begins now at an early age. Most schools have their own small computer on which mathematical problems can be worked out. Examination details go into a computer and the results when you get them will be in the form of a computer print-out. Even some examination papers – those having multiple choice questions – may actually be marked by a computer.

It is, in fact, very difficult to get away from the influence of computers. Even as you read this book you are seeing type that has been set with the aid of a computer, not to mention the many other computers that have been involved in its production, from the manufacture of the paper to the distribution of the printed book.

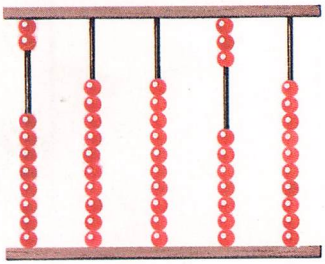


Computer Development

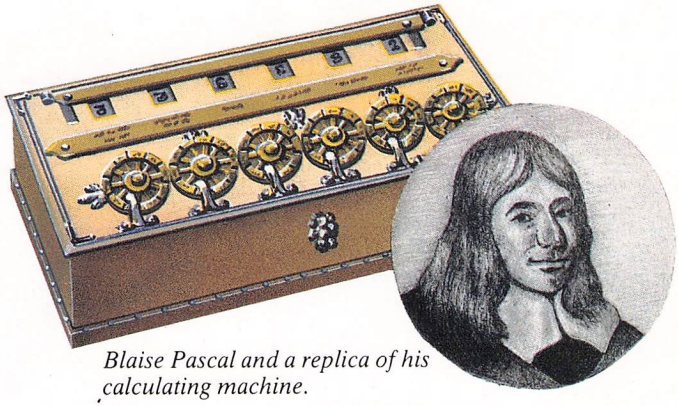
Perhaps the earliest device for working out sums was the abacus. This began as a clay tablet into which grooves were cut. Pebbles were then placed in or taken away from these grooves to perform addition and subtraction. (Our word calculation comes from the Latin word calculus meaning 'pebble'.) Because the pebbles were likely to become mislaid, they were later replaced by beads threaded on to wires and mounted in a frame. By moving the beads backwards and forwards, addition, subtraction, division and multiplication could be done.

In 1614 John Napier, an astronomer, invented a ready-reckoner, known as Napier's Bones, to help him make complex calculations accurately. From this was developed in 1621 the earliest form of slide rule.

The first real mechanical calculating machine, working with wheels, gears and dials, was made by a Frenchman, Blaise Pascal, in 1642. This machine used a number of wheels mounted parallel to each other, with 10 teeth mounted round the circumference of each. A carry-over mechanism was incorporated so that when one wheel made a complete revolution, the wheel to its immediate left was



The abacus was invented over 1500 years ago. A modern version (left) uses beads strung on wires. By moving the beads forwards and backwards, all arithmetic processes can be carried out.

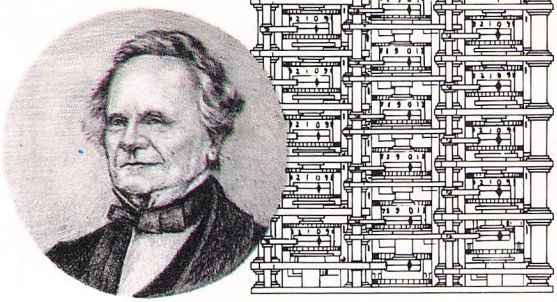


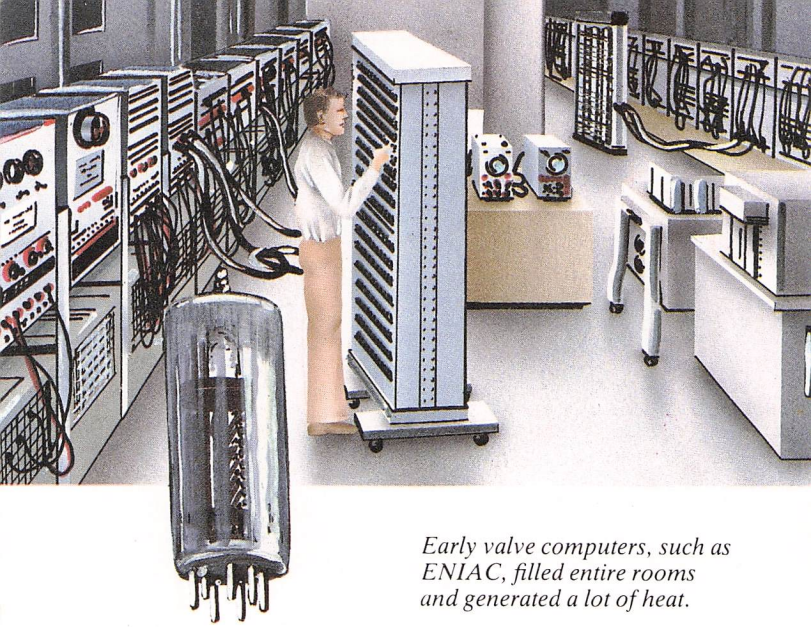
Blaise Pascal and a replica of his calculating machine.

turned through one segment – representing one '10'. The machine was operated by turning the wheels backwards and forwards, thus performing addition and subtraction.

The design for the first real computer was drawn up by an Englishman, Charles Babbage, in 1832. This was another mechanical machine but, like today's computers, it was designed to work automatically from a series of program instructions and also had the capacity to store the information on which it was working – up to 1000 groups of 50 decimal digits. Unfortunately, the technology of the

Sir Charles Babbage and his design for an 'analytical machine'.





Early valve computers, such as ENIAC, filled entire rooms and generated a lot of heat.

day was insufficiently advanced to manufacture the parts required for the machine and so it was never made.

It was not until 1944 that the first working automatic computer was built. This was at Harvard University in America and was known as an Automatic Sequence Controlled Calculator (ASCC). It was still, however, basically a mechanical device, rather like the one Babbage had designed, and therefore comparatively slow, taking about 4 seconds to perform a multiplication problem.

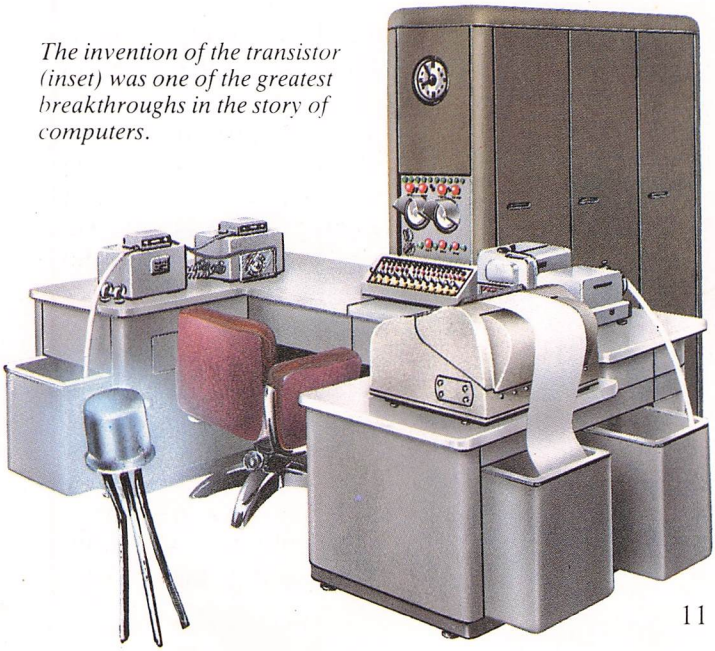
To develop a computer that would work really fast it was necessary to get away completely from the mechanical operation that had been used until now and have a machine, without moving parts, that worked in terms of electric pulses flowing through it.

The first automatic electronic computer was completed in Pennsylvania in 1946 and known as ENIAC

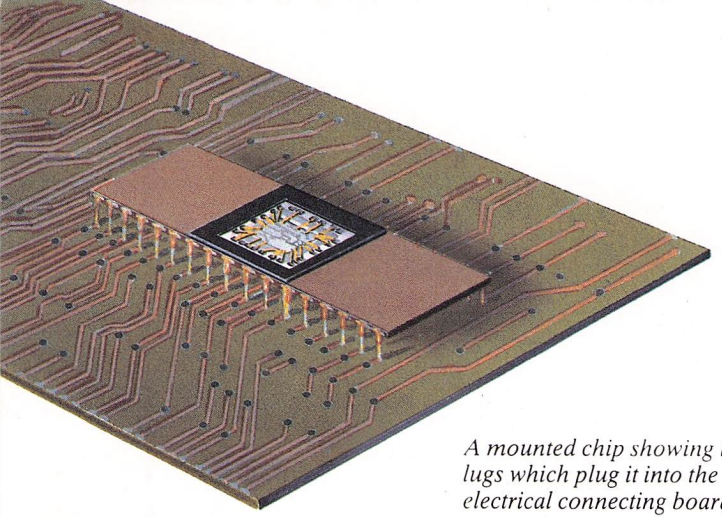
(Electronic Numerical Integrator and Calculator). It was a massive machine containing some 18,000 thermionic valves and weighing some 30 tonnes. Since then, many inventions, innovations and developments have been applied to this basic principle, resulting in the extremely powerful and fast machines in current use.

One of the more important developments was the introduction of devices that would record and hold information magnetically on magnetic tapes, discs and drums. A system was also developed of holding numbers magnetically in a series of small magnetizable rings.

A second very important development was the invention, in 1948, of transistors. These were first used in computers in the early 1960s to replace the



The invention of the transistor (inset) was one of the greatest breakthroughs in the story of computers.



A mounted chip showing the lugs which plug it into the electrical connecting board.

thermionic valves and diodes used in the earlier machines. This resulted in a considerable reduction in the size and a great increase in the power of computers.

The Silicon Chip

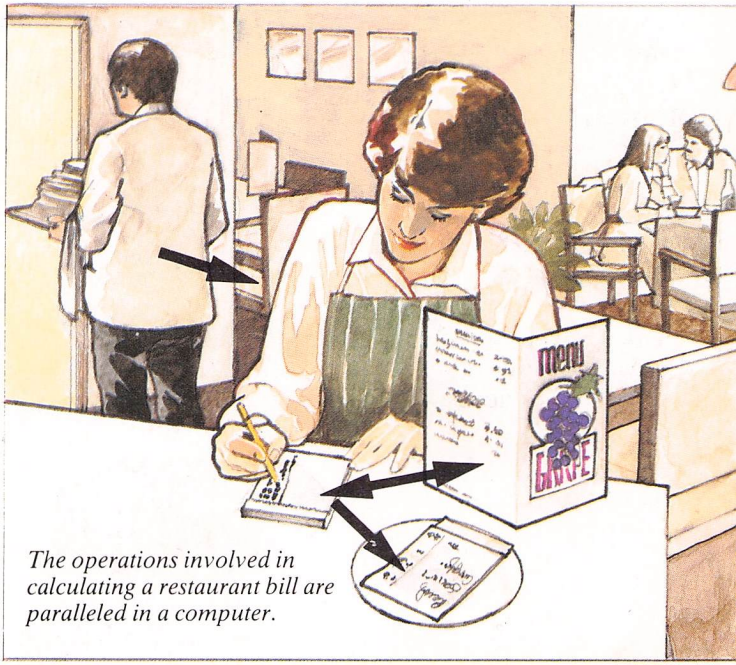
Today's computer is a far cry from the machine of the early 1950s. The development of micro-electronics resulting in the silicon chip (a very small device which may contain many thousands of transistors) has led to the introduction of machines known as microcomputers. Today, the computing power contained on a single silicon chip only 5 mm square and 0.1 mm thick maybe far greater than that of the first computer weighing 30 tonnes with its 18,000 valves; a microcomputer, purchased for less than £50, can be more powerful than the early machines costing as much as £1,000,000; and a modern machine may well perform a million simple calculations in a second rather than taking 4 seconds to do one like the first machine.

The Elements of a Computer

Computers vary tremendously in size, from desk-top models to huge machines occupying entire buildings, but they all contain the same component parts which come under the heading of *hardware* and *software*. Hardware is the term given to all the machines that make up a computer and the collection of machines is called a *computer configuration*. Software refers to the programs that instruct the machine.

To illustrate how a computer works, imagine the operations carried out by a waiter in a restaurant.

1. He accepts information from a group of diners – their order for the meal – and notes it on his pad. It is this information that will determine what they eat and what they will eventually pay. This could be called the *data* relating to the meal that is entered, or *input*, into the ordering system.
2. The waiter, before preparing the bill, will have to find out the cost of each dish ordered. To do this he may refer to the menu, read from this the prices and note them down on his pad. In effect he is extracting information permanently *stored* on the menu and then temporarily storing the prices on his pad until the bill is prepared.
3. In order to work out the bill the waiter will multiply the cost of each dish by the number served, note down the answers on his pad and then add them all up to get the total cost of the meal. This can be called the *arithmetic* of the system. Because the waiter is relating all the items of data on his pad in



The operations involved in calculating a restaurant bill are paralleled in a computer.

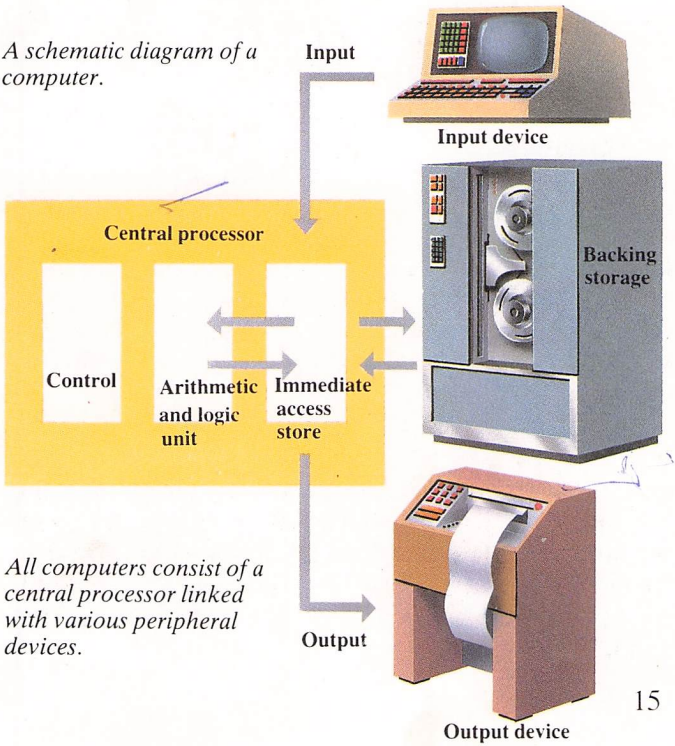
order to provide an answer the whole process can be called *data processing*.

4. Finally, when the diners are ready to leave, the waiter will present them with the bill which is the *output* of his ordering system.

In doing this, the waiter is working to a *program* that consists of a list of *instructions*. These instructions must be carried out in the right sequence if the required result, an accurate bill, is to be obtained. As far as the diners are concerned they will want to check that the bill has been accurately prepared. This involves imposing a degree of *control* over the situation.

Now, in principle, a computer works in much the same way as the waiter. It is concerned in accepting data, storing all of the information that is relevant to it, processing the data in a pre-determined way that conforms with a program of instructions, and then communicating the results as an output to the system. In fact, all of the factors mentioned above must be present in a computer and there must be machines, or devices, to carry them out as follows:

Central processor This is the machine that does all the working out that the waiter did on his pad. It has an immediate access store (IAS) for holding the data



A schematic diagram of a computer.

All computers consist of a central processor linked with various peripheral devices.

required for immediate use (sometimes this is called the computer's *memory*). It has an arithmetic and logic unit (ALU) which performs all of the calculations. It has a control element which monitors everything that is going on and draws attention to any errors that occur, and it holds in its immediate access store the program of instructions necessary for carrying out the job.

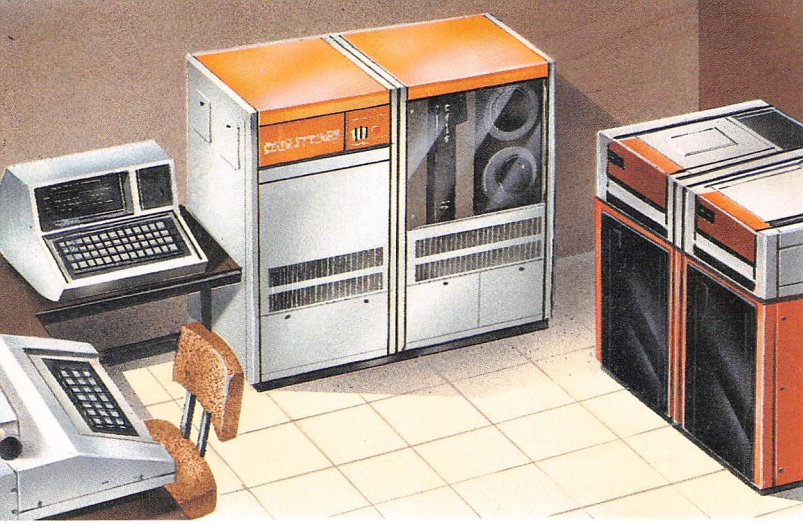
Input devices These machines are for reading and transferring the input data to the central processor, rather as the waiter read and transferred the prices to his pad.

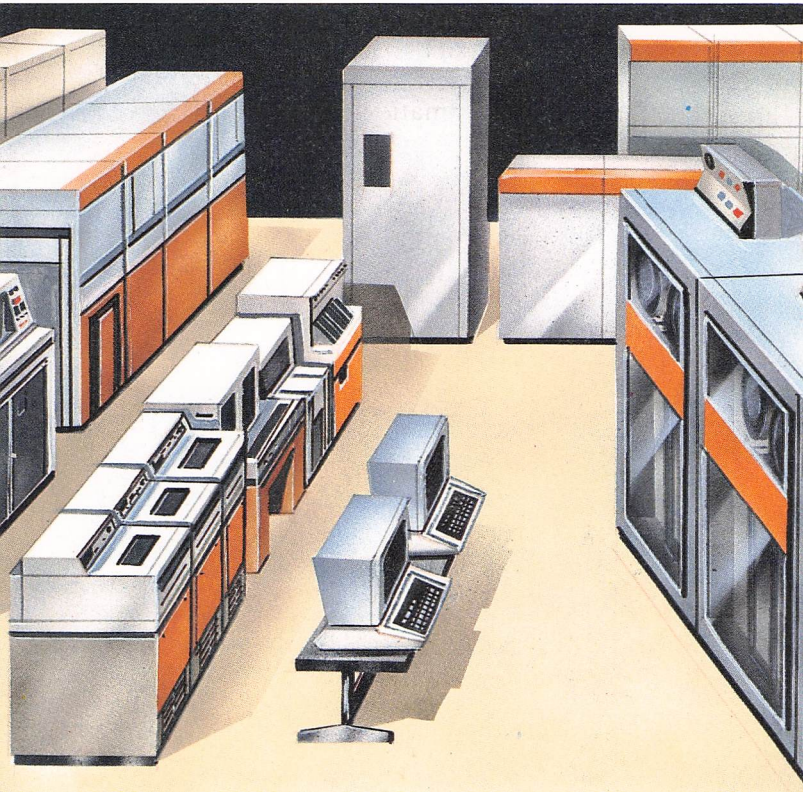
Output devices These are machines to which the central processor transfers the results of the processing, where they are printed out or displayed, rather like the bill worked out in the restaurant.

Backing storage devices These are the devices that hold the mass of information that may be transferred for use during processing, as and when required. The restaurant menu is an example of this.

A computer, then, really consists of a number of machines. The main one is the central processor and connected to it electrically are a number of other machines – called *peripherals* – for input, output and storage.

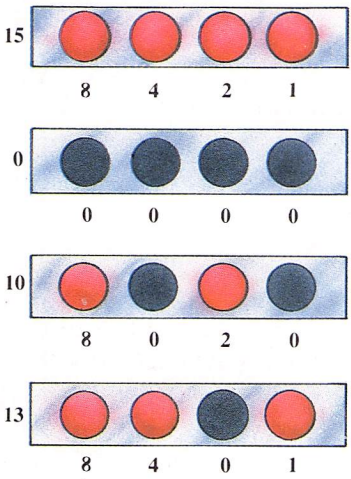
Computers vary greatly in size and capacity from the microcomputer (left), through the minicomputer (top right), to the powerful main-frame computer (right) but they all contain the same basic elements.





Communicating With a Computer

A computer can easily perform the billing operation carried out by the waiter in the restaurant or, indeed, tasks of far greater complexity. The problem is how to communicate to the computer the information it needs to perform the operation in a way it can recognise and understand. Unfortunately, a computer cannot store and work on all the wiggly lines and shapes that we recognise and understand as letters and numbers. A computer is an electronic device that works by electric pulses flowing through it. This means it can only recognise and understand two things: the presence of or the absence of such a



A panel of lamps can be used to represent a range of numbers. In the example shown (left) the four lamps represent 1, 2, 4 and 8 respectively when on and 0 when off. A combination of 'on' and 'off' lamps can represent any number between 0 and 15.

pulse. Everything communicated to the machine must, therefore, be expressed in these two terms.

When we communicate with each other there is a wide range of alternatives we can draw upon. They include 26 letters (A–Z), 10 digits (0–9) and a number of symbols ?, £, @, etc. But this is not the only way of communicating with each other; we could, for example, use a series of dots and dashes – the Morse Code – or a series of shorthand outlines. Provided you know what code is being used and can convert it to the standard characters normally used, then you can understand it.

The Lamp Code

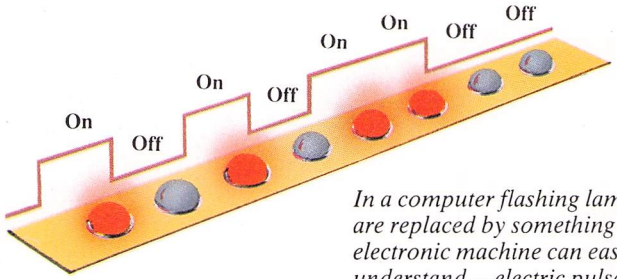
You could, for instance, communicate using an electric lamp by having a code that made 'Off' mean 'No' and 'On' mean Yes'. Or you could give Off and On numeric values, making Off equal 0 and On equal 1. If you added a second lamp for which Off equalled 0 and On equalled 2, you would be able to communicate four numbers by just switching the lamps on and off: Off-Off = 0, Off-On = 1, On-Off = 2, On-On = 3. Add a third lamp to represent Off = 0 and On = 4 and you can represent a range of 8 numbers (0–7), while a fourth lamp, Off = 0 and On = 8, increases the range to 16 (0–15).

The Binary System

What you are now doing is using a number system based on two alternatives only, Off and On. This is in contrast to the decimal number system we normally use that has ten alternatives, 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. A number system using just the two digits 0 and 1, is known as a binary system and is ideal for computer use in which the absence of an electric

Traffic lights are an example of a binary situation. Each lamp can be either on or off. Traffic lights are also an example of a relatively simple computer-controlled operation.

pulse is made to represent 0 and the presence of a pulse 1. Instead of having four electric lamps to show the numbers 13 (On-On-Off-On) we pass an equivalent string of pulses through the computer . This could be written as 1101 – the binary equivalent of 13.



In a computer flashing lamps are replaced by something an electronic machine can easily understand – electric pulses.

The Binary Code

The important thing to remember about the binary number system is that it has a base of 2 (this is called the radix). A decimal system has a base (radix) of 10. In a decimal system, every time 10 is reached a carry-over occurs by moving a 1 to the left followed by a zero. In a binary system, this carry-over occurs every time 2 is reached and is shown in the same way by moving a 1 to the left followed by a zero.

In decimal $6 + 4 = 10$ (decimal 10)
In binary $1 + 1 = 10$ (decimal equivalent 2)
and $1 + 1 + 1 = 11$ (decimal equivalent 3)

In a decimal system, place values increase by an additional power of 10 moving from right to left of an expression; in a binary system, these place values increase by additional powers of 2.

Place values in a decimal system are 10^4 10^3 10^2 10^1 10^0 which means the decimal number 23,456 equals:

$(2\times10^4)+(3\times10^3)+(4\times10^2)+(5\times10^1)+(6\times10^0)$
$= 20{,}000 + 3000 + 400 + 50 + 6 = 23{,}456$

Place values in a binary system are 2^4 2^3 2^2 2^1 2^0 which means the binary number 10111 equals:

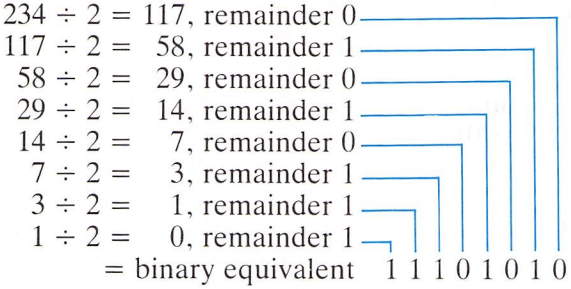
$(1\times2^4)+(0\times2^3)+(1\times2^2)+(1\times2^1)+(1\times2^0)$
$= 16 + 0 + 4 + 2 + 1$ = decimal equivalent 23.

The fact that a computer uses the binary notation to store and process all of the information it deals with does not mean that conversion from decimal to binary has to be done manually before entry to the computer. Information is entered through an input device in ordinary alphabetic and numeric characters and the computer itself then converts it all to binary. Nevertheless, the principle of decimal binary conversion is quite simple,

Converting Decimal to Binary

Divide the decimal number successively by two until it is reduced to zero. When, on division by two there is a remainder of one, this becomes a binary digit 1; if there is no remainder it becomes a binary 0. The binary expression is built up from right to left.

Example: To convert decimal number 234 to binary.


234 ÷ 2 = 117, remainder 0
117 ÷ 2 = 58, remainder 1
58 ÷ 2 = 29, remainder 0
29 ÷ 2 = 14, remainder 1
14 ÷ 2 = 7, remainder 0
7 ÷ 2 = 3, remainder 1
3 ÷ 2 = 1, remainder 1
1 ÷ 2 = 0, remainder 1
= binary equivalent 1 1 1 0 1 0 1 0

Converting Binary to Decimal

The decimal value of a binary expression is equal to the sum of the decimal values of the binary digits.

Example: To convert binary number 101011 to decimal –

$$\begin{array}{cccccccccccc} & 1 & & 0 & & 1 & & 0 & & 1 & & 1 \\ = & 2^5 & + & 0 & + & 2^3 & + & 0 & + & 2^1 & + & 2^0 \\ = & 32 & + & 0 & + & 8 & + & 0 & + & 2 & + & 1 \end{array} \quad = \text{decimal } 43$$

Characters and Symbols

The computer is not only concerned with the expression of numbers in binary. It will also have to deal with information expressed in alphabetical characters A–Z and also with a number of symbols e.g. @, ", /. These also have to be converted into binary numeric coding. This can be done by representing each character by a unique binary expression that the

computer will recognise as meaning the character indicated. For example, we can divide the 26 alphabetic characters into three groups of nine, nine and eight, consisting of A to I, J to R and S to Z respectively. Four binary digits can be used to indicate the place of the letters within the group, for instance A, the first letter, becomes 0001, and I, the ninth letter, becomes 1001. These groups of four are prefixed by two additional digits to indicate the group within which each letter falls, say 01 to represent the first group, A–I, 10 for the second, J–R, and 11 for the third, S–Z. This means that A is represented by 010001 and Y by 110111.

When converting numbers into binary we dealt with the number as a whole and finished up with a single binary expression representing the decimal number. This is known as *pure*, or *serial*, binary. There is, however, another way of expressing numbers in binary that is extensively used in computers. This is to show each individual decimal digit as a separate binary expression.

The range of binary expressions used for the 10 decimal digits are:

Decimal	Binary	Decimal	Binary
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001
5	0101	0	0000

The decimal number 759 would, therefore, become:

0111 0101 1001

This method of representing decimal numbers in binary is known as *binary coded decimal*.

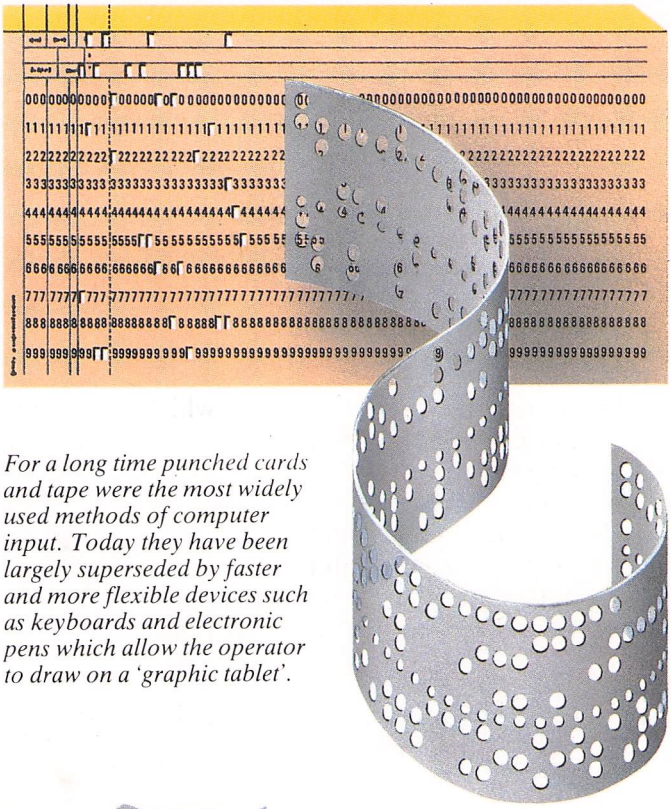
Inputs and Outputs

In earlier days computers were sometimes known as 'electronic brains'. This is rather an exaggeration as, while a computer may work a lot faster than we can, it certainly does not have the creative thinking power or the perception of the human brain. In fact, it can only work in the simplest of terms – on and off, or zero and one. However, the two are similar in that they both work from information supplied, or input, to them, and they are both able to communicate the results of their working.

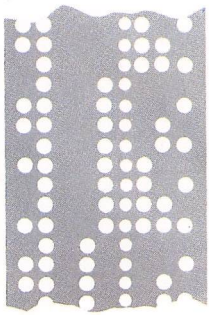
There are a number of ways in which the human brain can do these two things. We take in information by sight, by hearing, by touch, etc., and we can communicate by writing, in speech, by gestures, etc. Similarly, there are a number of ways information can be input to a computer and a number by which its output can be communicated via devices connected electrically to the central processor unit.

Computer Input

One way to input information is to record it first on a document of one type or another, then pass the document through an input device which will read and pass the information on to the processor as strings of binary pulses. Punched cards and punched paper tape are often used for this purpose. In both cases the information is recorded by punching holes, either in cards or in a continuous strip of paper tape. In the case of cards, each position on the card at which a hole may be punched is given a predetermined value or meaning, say a digit or a letter, so that the reading device, on sensing the hole, will know its meaning from its position on the card. In



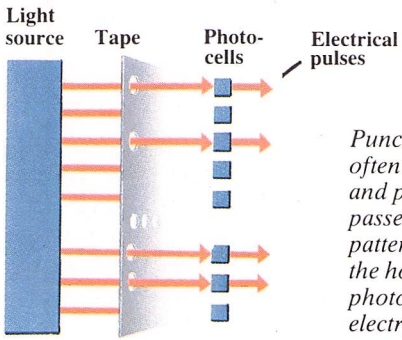
For a long time punched cards and tape were the most widely used methods of computer input. Today they have been largely superseded by faster and more flexible devices such as keyboards and electronic pens which allow the operator to draw on a 'graphic tablet'.



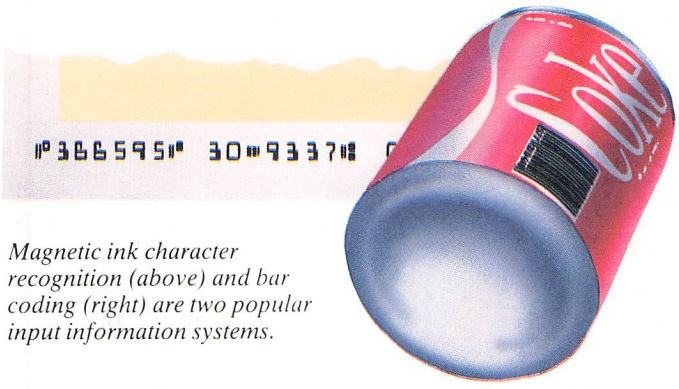
The pattern of holes in punched cards and tape represent characters which can be recognised by the computer's reading device. The piece of tape (left) contains the code for individual letters of the alphabet. Tape can be read at about 1000 characters per second.

the case of paper tape, the character is represented by a pattern of holes punched across the tape in the binary code that represents the character. Until recently punched cards and paper tape were used extensively but they are now rapidly losing their popularity in the face of more convenient and less laborious input methods.

In character printed documents the information is printed in ordinary numbers and letters but in a special stylized form that enables the reading device, or *reader*, to recognise them. One method, known as magnetic ink character recognition, uses a special ink that can be magnetized so that individual characters can be recognised by the reader from the shape of the wave-form of the electric pulse induced by the magnetism. Magnetic ink character recognition is used by banks and you will find magnetic ink characters printed along the bottom of cheques. Another method, known as optical character recognition, uses ordinary ink: the shape of the character is recognised by the reader scrutinising it with a light source in conjunction with photo-electric cells. Optical characters are used extensively in preparing gas and electricity bills.



Punched tape reading devices often consist of a light source and photocells. As the tape passes through the reader the pattern of light shining through the holes is converted by the photocells into a pattern of electrical pulses.



Magnetic ink character recognition (above) and bar coding (right) are two popular input information systems.

A third way is to print the binary coding representing a character as a series of lines, or bars, on a document. This is known as bar coding and you will find examples of it on many products you buy from shops. Another way to input information is to record it magnetically on a disc or on tape.

In all of the above methods input is really a two-stage affair. First the information is recorded onto a document or a magnetic medium and then it is read through an input device for transfer to the processor. However, there are ways of transferring information direct to the processor, in one stage. The most commonly used method is through keyboard machines connected direct to the processor.

Some of these are rather like the ordinary typewriter keyboard. On depression of a key, the binary code representing the character is automatically generated and transferred to the processor. Others are simple keypads which need only be touched in order to record the information. You will find an example of this in the cash-dispensing machines to be found outside banks. Many of these direct input keyboard devices incorporate a visual display.

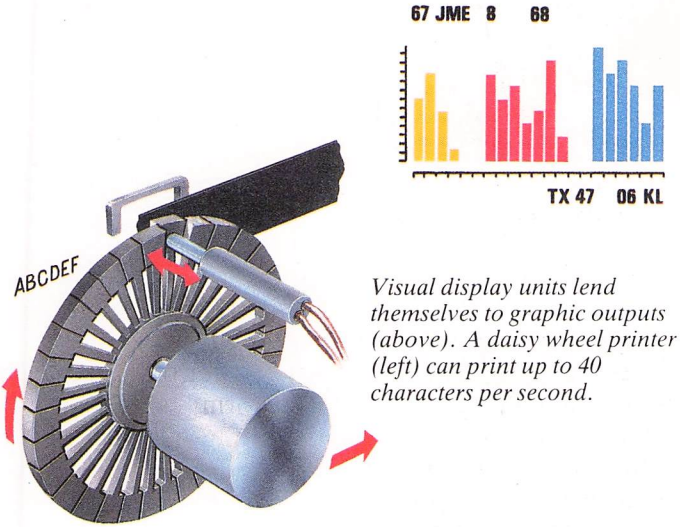
A modern, touch-sensitive keyboard computer.

Computer Output

There are two main types of computer output: one is a permanent record in printed form; the other is a temporary display on a visual display unit (VDU). There are two main kinds of printed output,often known as impact printing and non-impact printing.

Impact printing is when type is impressed on paper. Some printers, known as line printers, will print a whole line of type up to, say, 120 characters long in one go. They are very fast and will print up to 400 lines per minute. Other printers, known as serial printers, are very much slower, printing just one character at a time. The characters may be carried on ordinary type bars, like those of a typewriter, or on a wheel. This revolves until the correct character is in the printing position and is impressed on to the paper through a carbon ribbon. This 'daisy wheel' printer will print up to about 40 characters per second, while the typewriter kind of printer will only achieve about 10.

Non-impact printers, as the name suggests, are those that do not make an impression on paper. They include electrostatic printers which operate in principle very much like photo-copying machines, and printers which reproduce printed information



Visual display units lend themselves to graphic outputs (above). A daisy wheel printer (left) can print up to 40 characters per second.

photographically. The photographic record takes two forms, known as microfilm and microfiche. Microfilm uses rolls of 16 mm film recording a page of information on each frame, while microfiche is a sheet of film, say 100 mm by 75 mm, on which up to 80 pages of information can be recorded. One major advantage of this photographic means of output is its speed – up to 100,000 characters per second. One disadvantage is that special equipment is required to project a magnification of the image in order for it to be read.

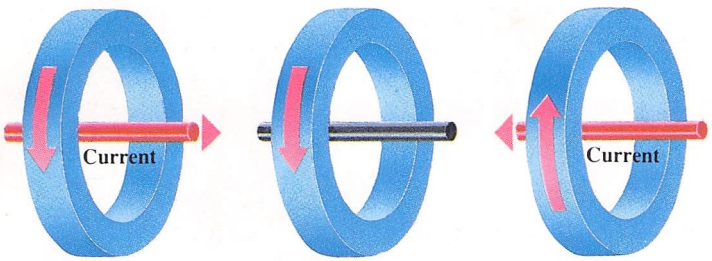
When a permanent record is not required, output may be displayed on a VDU. This is particularly useful for just 'looking-up' a record held on the computer, a request for the information being keyed into a keyboard attached to the VDU.

The Central Processor

The central processor unit (CPU) is the 'brain' of the computer. This is where all the work is carried out. The main functions of the central processor are: to store the information on which it is working; to store the program of instructions it has to carry out; to carry out the instructions; and to exercise control over all of its activities to make sure everything is being done correctly and accurately.

All of the information held in the central processor is available for immediate use and this store is known as an immediate access store. It does not contain all the information used by a computer. For example, a computer processing all of the motor vehicle records in the country would be involved in such a vast mass of information that the whole of it could not possibly be stored in the processor itself. Instead, it is stored in backing storage – magnetic tapes or discs – that is connected to the processor. The processor is able to extract any particular item of information required

Ferrite cores are still widely used in computer memories. The core can be magnetized by an electric current passing through it. The magnetic state, or polarity, depends upon the direction of the current and so can be used to store a binary character.



Magnetic state (polarity) **Polarity remains** **Polarity changes**

immediately to be worked on and to return the information to the backing storage when it is finished with.

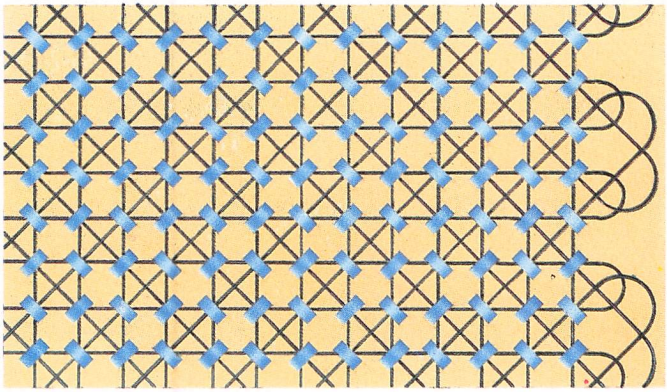
Since all of this information has to be held in terms of 0 and 1, the store must be made up of a lot of devices, each one capable of representing either of these two variables, each one capable individually of being changed from 0 to 1 or from 1 to 0 as different information is required to be stored, and each one capable of being 'read' when required to ascertain whether it is in an 0 or a 1 state.

Ferrite Cores

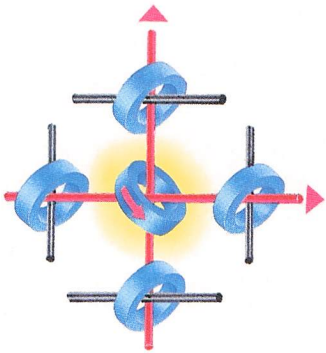
Until recently a device known as a ferrite core was used in most computers. Although this has now been superseded by semi-conductor devices, ferrite cores are still the basic memory element in most large computers in use today.

A ferrite core is a small ring, about 1 mm in diameter, capable of being magnetized in one of two directions, clockwise or anti-clockwise. It can be magnetized by passing an electric current along a wire threaded through the centre of the ring. If the direction of the current is reversed, the polarity of the magnetism is also reversed. Since the binary code for each character consists of a number, say six 0's and 1's, then a corresponding number of cores is necessary to hold each character. If we are to store not only a quantity of information but also the program necessary to process it, tens of thousands of these cores are required, each individual capable of being changed on demand.

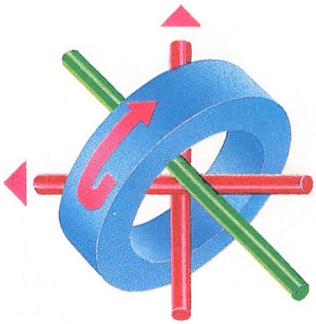
To do this, a large number of cores are arranged in a matrix and each line of cores threaded by wires in two directions at right angles to each other, so that



Ferrite cores are arranged in a matrix (above) and the matrices are built one on top of the other to provide a sufficiently large memory for a central processor.



Each core has two wires threaded through it at right angles (right). Half the current needed to switch the core is passed through each wire so that an individual core can be switched without others being affected.



A third 'sense' wire detects the state of a core.

only one core lies at the intersection of any two wires. If we pass a current through the centre of a core, its polarity will be changed, but the current must be of a certain minimum strength. If, instead, we pass currents of *half* this strength through two of the wires at right angles to each other, then the strength of current required to change the polarity of a core will only appear at the point where the two wires intersect. In this way any single individual core can be changed at will without any other core being affected.

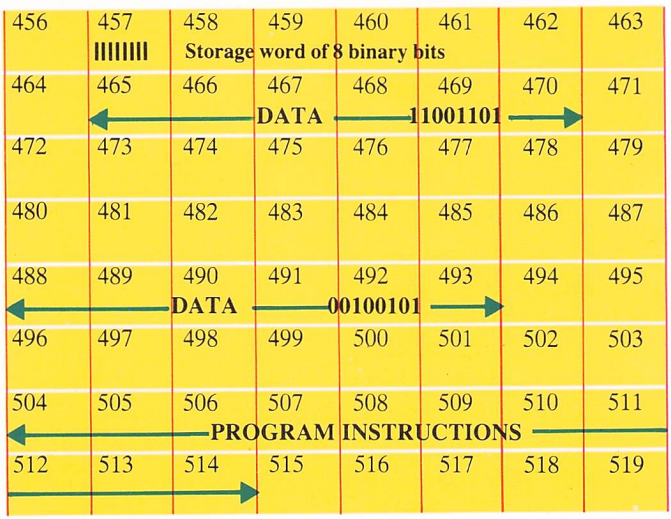
It is, of course, necessary to be able to 'read' what is contained in each ferrite ring. This is done by threading a third wire through them which will, on command, sense the state of each ring. This is known as a *sense wire*.

Words and Addresses

Since all of the information is stored in this mass of cores representing 0's and 1's, we need to know where one group of cores representing a specific character (a letter or a digit) begins and ends; we also need to know where the character is located so that it can be found again when required. The answer to the first problem is to divide all the cores up into groups of a standard size. The size of the group depends on the type of computer, it may be 6, 12, 18, 24 and so on, but the important thing is that each group will contain a pre-determined volume of information. For example, if there were six cores in a group then this would contain just one character; if there were 24 it could contain 4 characters. This division into groups is made by mounting a number of core matrices vertically on top of each other. Each vertical column of cores then represents one group.

The name given to such a group is a *word*. The answer to the second problem is that each word is given a unique reference number, which is known as the *address* of the word. This means we can always locate it by quoting the address of the word. We do not have to keep this record ourselves; the computer does it for us by indexing the contents of each word against the word's address. These storage locations, or words, are used both for storing the information to be worked on and also the program instructions that have to be carried out. It does not matter where, or in what order, the information is stored (the

A schematic diagram of a central processor store. Words 465–470 and 488–493 contain data items. Words 504–514 contain program instructions. Word 469 contains the binary equivalent of 205. Word 492 contains the binary code for the letter N.



computer itself will keep a note of the address at which each item is located) but the program instructions must be held in a series of consecutive addresses so that, having executed the first instruction, the computer knows that the next instruction is to be found in the next address.

Semi-conductors

In modern computers semi-conductors are used as the basic storage element rather than ferrite cores. However, the store is organized in much the same way, each semi-conductor being capable of representing one binary digit, or *bit*, and a standard number being grouped together to represent a computer word. A semi-conductor is simply a device that will permit an electric current to pass in one direction only, a property possessed by silicon after it has been suitably treated. With 'wires' attached either end of the silicon, known as *source* and *drain* wires, a current will flow from source to drain if a voltage is applied across them. The conductivity of the silicon (that is, the current flow) can, however, be controlled by placing a further electrode, known as a *gate*, between the source and drain and applying a voltage to that. This creates a two-state situation where a high degree of conductivity can represent a binary 0 and a low state a binary 1. Moreover, that state can be controlled by the voltage applied to the gate. The binary representation of the device can be found by testing the current flow through the source wire.

Semi-conductor memories have the advantage that they are very cheap to produce compared with ferrite cores, they operate with tiny electric currents, and many thousands of semi-conductors can be packed together in a very small space.

Computer Arithmetic

One of the main functions of a computer is to carry out mathematical operations at great speed. Any arithmetic problem, however complex, can be reduced to the four basic arithmetic rules of addition, subtraction, division and multiplication. A computer works so fast that we can use this great speed to simplify its arithmetic even further. Multiplication can be carried out by repeated addition: if we want to multiply 76 by 85 we can do so by adding 76 to itself 85 times. Division can be carried out by repeated subtraction: if we want to divide 125 by 25 we can see how many times we take 25 from 125 before zero is reached. This leaves just addition and subtraction. And subtraction can be carried out by a process known as complementary addition.

In this way all of the arithmetic processes in a computer can be reduced to the simple addition of binary numbers, with the following four basic rules:

0 + 0 = 0
0 + 1 = 1
1 + 1 = 0 carry 1
1 + 1 + carry 1 = 1 carry 1

The section of the central processor in which the arithmetic is carried out is known as the arithmetic and logic unit. This contains a number of special registers and electronic circuits known as *gates* through which the binary statements on which arithmetic is to be performed are passed. As can be seen from the above table, the number of different operations required is quite small and they are also, in arithmetic terms, very simple. These operations are known as *logical operations*.

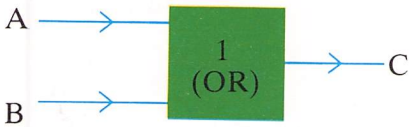
Logical operations

The simplest logical operation is that which reverses the value of the binary digits and is accomplished through a device known as a NOT gate. If an 0 is input, 1 is output; if 1 is input, 0 is output.

A two input OR gate will produce an output pulse as a result of the application of one pulse to either input or simultaneously to both.

Logical OR Table

Input A	Input B	Output C
0	0	0
1	0	1
0	1	1
1	1	1

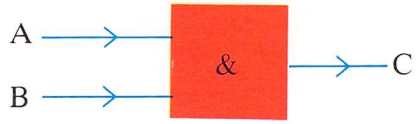


C = 1 when A or B or both = 1

A two input AND gate will give an output pulse only if pulses are applied to the two inputs simultaneously.

Logical AND Table

Input A	Input B	Output C
0	0	0
1	0	0
0	1	0
1	1	1

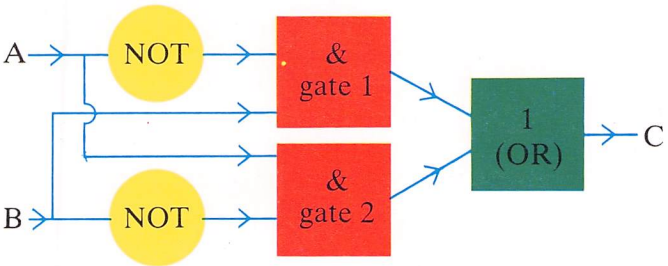


C = 1 only when both A and B = 1

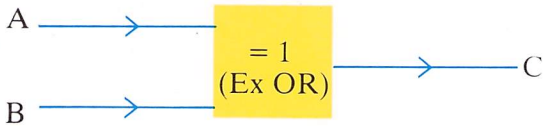
Using a combination of NOT, AND and OR gates we can construct what is known as an Exclusive OR gate (sometimes known as a half-adder) through which we can trace the four combinations of the values of A and B as follows:

Half-Adder

Input		Input AND gate 1		Input AND gate 2		Input OR gate		Output C
A	B							
0	0	1	0	0	1	0	0	0
1	0	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	1
1	1	0	1	1	0	0	0	0



An exclusive OR gate is shown like this. One input on A or B but not both gives one output.

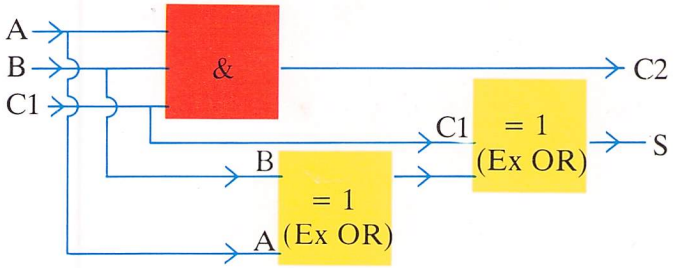


Full-Adder

You will note from the above table that an Exclusive OR gate meets the requirements of the four basic binary addition rules listed earlier with one exception; it does not give a carry over as, for example, in the case of 1 + 1 = 0 carry 1, or 1 + 1 + 1 = 1 carry 1. This carry-over, by definition, involves a second output pulse (C2) and a subsequent third input pulse (C1) as shown in the following table:

Inputs			Outputs	
C1	A	B	S	C2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Using a combination of exclusive OR and AND gates, we can devise circuitry to meet these requirements. This is a binary one-bit full-adder. The third (carry-over) pulse C1 becomes a third input to the AND gate which will give one output pulse C2 if either two or three inputs are present.



Inputs A and B are applied simultaneously to the AND and the first Exclusive OR gate, the output of the latter becoming the input to the second Exclusive OR. Any output from the AND gate (C2) is carried to the next set of gates as the carry-over (C1) and input simultaneously to the AND gate and the second Exclusive OR.

Of course, this adder will only accomplish the addition of one pair of binary bits. In a computer we have to add together the groups of bits representing numbers. This is done by arranging a number of these adders in parallel so that all the bits in groups can be dealt with simultaneously.

Program Instruction

Like any other machine, a computer is designed to perform a finite range of operations such as add, divide, shift, read, jump, etc. Each of these operations is identified by a unique binary code, say 1011 for add, 1100 for read, etc. The complete code is known as the computer's operation code or function code.

When a specific code is communicated to the arithmetic and logic unit, the computer knows what operation it has to perform. There is not much point, however, in just telling it to add; it must be told what to add. This means that the information on which the operation is to be performed has also to be identified.

A program instruction must, therefore, contain both of these elements – an operation code to show what is to be done, known as the *operation* element of an instruction, and an identification of the information to be worked on, known as the *operand* element of the instruction.

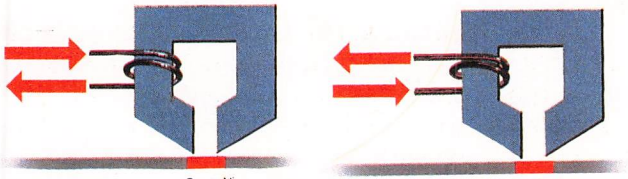
Information Storage

It is not possible to store all of the data a computer requires in the central processor itself, but additional stores, known as backing stores are used, to and from which the processor can transfer data as and when it is required. Two main storage devices are used for this purpose – magnetic tapes and magnetic discs.

Magnetic Tape

The tapes used in computers work on much the same principle as ordinary domestic tape recorders, except that they are usually on a much larger scale. The tape itself is a continuous length of plastic material some 125 mm wide and up to 750 m long that is coated with a magnetizable surface. The tape is wound on to a spool that can be loaded on to a tape-deck as and when required. The deck is connected permanently to the central processor and information is written to or read from the tape through the processor on the

Magnetic tape is used to store binary digits. The magnetic state, or polarity, of a particular spot on the tape depends upon the current direction in the write head. Reversing the current reverses the polarity. In this way a binary 1 or a binary 0 can be written.



Binary 1 **Binary 0**

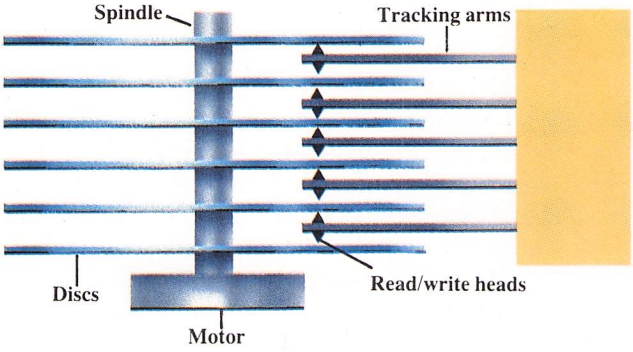
command of its program. As we have already seen, all characters can be expressed in a binary coding and it is this coding that is used to record them on the tape by magnetizing a pattern of dots across its width. Of course, to make sense of what is recorded the groups of characters making up words, numbers, etc. have to be separated from each other. This is done by inserting between them special patterns of bits, known as markers, that the computer will recognise as marking the end of a word, a sentence, etc.

Very large quantities of information can be recorded on a tape, say up to about 20 million characters on 750 m, and information can be transferred to and from it at very fast rates, often in excess of 200,000 characters per second. One major disadvantage of this form of storage is that, if a specific item of information is required the tape has to be run through with every item being scrutinized until it is found. For this reason tapes are only used when the information is to be processed in the same order as it appears on the tape. This is why it is referred to as a serial access form of storage.

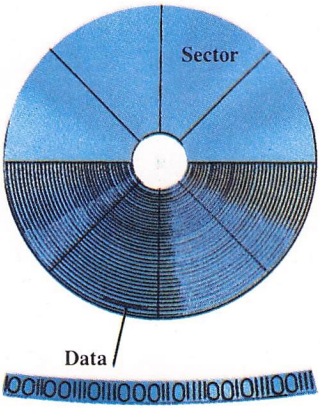
With the new generation of microcomputers small magnetic tape systems have been introduced that are virtually identical to the cassette tape systems used for sound recording.

Magnetic Discs

In contrast to a magnetic tape, backing storage on a magnetic disk makes it possible for any selected item of data to be found, or *accessed*, directly, without reference to the items stored before or after it. For this reason it is referred to as random access, or direct access, storage.



A schematic diagram of a disc unit (above) showing how the tracking arms can gain access to information on any part of the disc surfaces.



The disc surfaces are divided into sectors and each track within a sector has its own reference number (left).

Although there are a number of different disc systems available, the most commonly used in large computer installations, known as a disc pack, consists of six discs mounted parallel to each other on a central spindle. The disc pack can be mounted or removed at will from a disc transport that is permanently connected to the central processor. This system gives a total of ten recording surfaces in each pack (the top and bottom surfaces are not used for recording).

Each surface is divided into a number of concentric recording tracks, say 200, and each track in turn

divided into a number of sectors (usually 8), each sector capable of recording a fixed maximum number of characters in binary coding as small magnetized dots. Information is written to, and read from, the disc surface through read/write heads, one for each surface, that can be moved across the surface of the disc in order to locate any required track. The disc rotates at high speed, so that each sector of the track passes under the head in turn.

Since the object of a direct access storage system is to enable any data item to be accessed at will, there must be some system of locating where each data item is stored. This is done by giving each sector on each track on each surface a unique reference number, for example 091567 could be the reference for the 7th sector of track number 156 on surface number 9. In addition, each data item also incorporates a unique reference, known as the data record key. If the two references are indexed (a task accomplished automatically by the computer itself), the location address can be generated by reference to the data record key.

While the volume of data that can be held on a disc pack varies from model to model, some packs hold as many as 200 million characters (that is the equivalent of the text contained in more than 3000 books of this size). Characters can be transferred to and from the disc at up to about 800,000 per second, while the time it takes to locate any one item stored on the pack is around 40 milli-seconds.

Small flexible discs have been developed for use with microcomputer systems. In principle these work in much the same way, except that they are single discs rather than a number assembled into a pack. They are known as floppy discs.

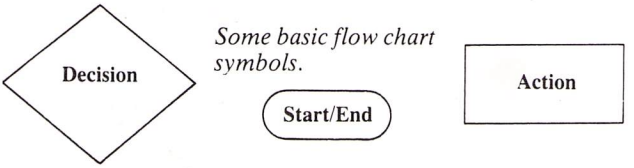
Computer Programs

A computer program is a list of instructions, arranged in a logical sequence, which when followed achieves a pre-determined aim, taking into account all of the alternative situations that may arise in pursuing the aim.

Let us say, for example, a man has just come home from work and his aim is to get inside his house. This seems easy enough on the surface – he just opens the door and walks in. But what if he finds the door is locked? What does he do now? Ring the bell; perhaps his wife is in and will open the door for him. No response? His wife must be out. He looks to see if he has a key in his pocket. No? Who has got a key? His wife has, and he knows she is either at the library or at their son's house. He decides he will try both places twice, and if he cannot find her return home and wait until she arrives. If he does find her, he will get the key, return home and let himself in.

Logic Flow Charts

While this perhaps is not the kind of problem we would normally use a computer to work out (usually the problems are mathematical or financial) we can see from this example how, in principle, a computer program is constructed. First of all we work out the logic of the situation to show what actions must be taken resulting from the various situations the problem may give rise to. This is usually worked out in chart form, known as a logic flow chart, which consists of a number of symbols with the actions written inside them and joined together by a number of flow-lines. In the flow chart illustrated, only three



Some basic flow chart symbols.

symbols are used: one for the start and the end of the problem; a rectangle to denote an action that must be taken; and a diamond to show where a decision has to be arrived at.

The chart illustrates a number of standard techniques used in flow-charting:

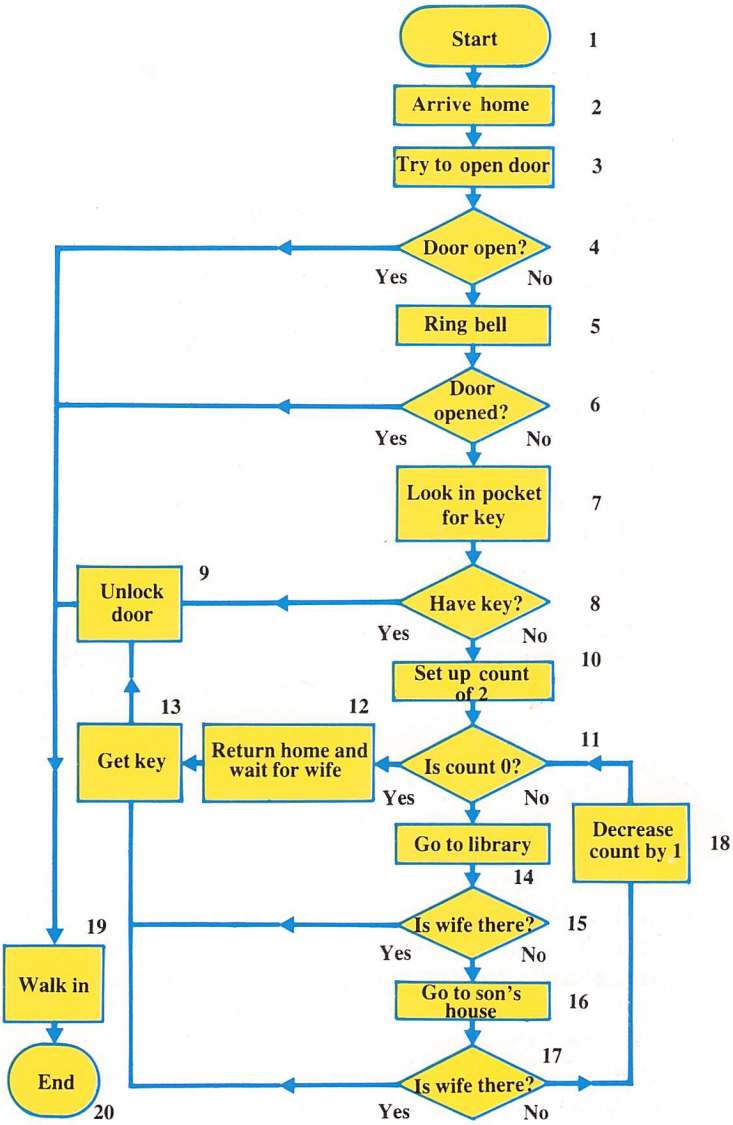
A Decision The first of these is symbol 4 from which flows two alternative courses of action arising from a decision as to whether the door is open or not.

A Loop This is set up when a group of instructions has to be repeated a number of times. In the chart these are the symbols 11, and 14–17. If a loop is set up the computer will continue to go round and round it for ever unless an escape from it is provided.

A Count This is when a register in the computer is used to count the number of times an operation is completed. In the diagram it is used to escape from the loop, having gone round it twice we go on to another action (12).

A Conditional Jump This is when the computer is directed to jump to another part of the program depending upon a certain condition being met. In the chart, if the answer at 11 is No, a jump is made to 14. If it is Yes we carry on to 12.

An Unconditional Jump This is when a jump is made from one part of the program to another without any condition. In the chart this happens at 13 when an automatic jump is made back to 9, ignoring the instructions in between or those that follow.


Start
1
Arrive home
2
Try to open door
3
Door open?
4
Yes
No
Ring bell
5
Door opened?
6
Yes
No
Look in pocket for key
7
Have key?
8
Yes
No
9
Unlock door
10
Set up count of 2
11
Is count 0?
Yes
No
12
Return home and wait for wife
13
Get key
18
Decrease count by 1
Go to library
14
Is wife there?
15
Yes
No
19
Walk in
Go to son's house
16
17
Is wife there?
Yes
No
End
20

Having charted the logic of the problem the next step is to write out a list of the instructions in a formal way that conforms with the rules of the particular programming language we are going to use. If we write out some of the instructions from the chart they could go like this:

```
 7 Look in pockets for key.
 8 If no key go to 10
 9 Unlock Door
10 Set up count of 2
11 If count greater than 0 go to 14
12 Go back and wait for wife
13 Get key
14 Go to library
15 If wife there go to 13
16 Go to son's house
17 If wife there go to 13
```

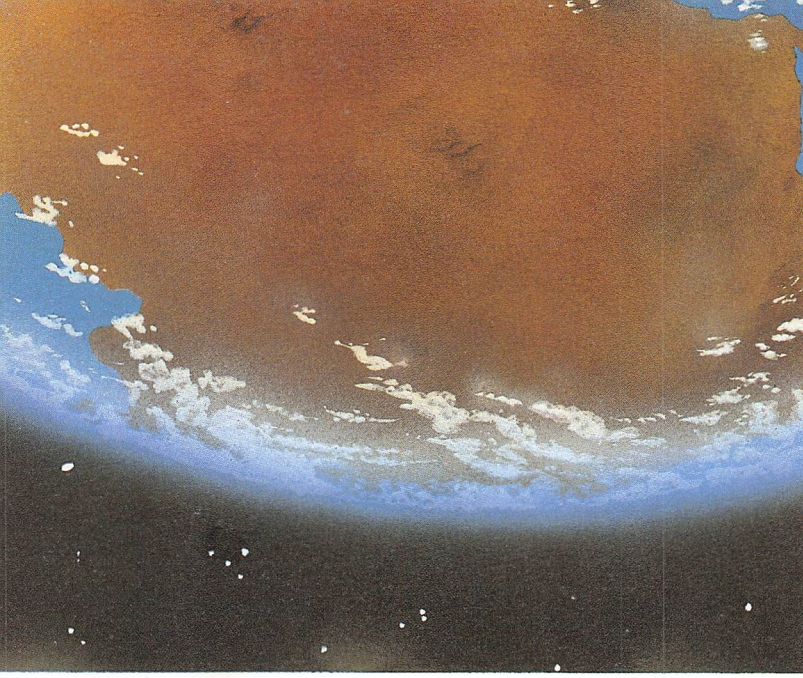
Computer Language

There are a number of different programming languages in which these statements can be written and the choice will depend upon a number of factors, such as the type of problem we wish to solve, the type of computer we are using, and the way in which we are going to use it. But in the final analysis the computer can only understand and execute instructions that are held in its store, expressed in binary code with each quoting an operation (or function) and a data address. Now some languages follow this pattern (it is called the machine instruction format) quite closely. For example, if the binary function code for multiply is 1100 and the address of a factor we wish to multiply is 1001, then the machine instruction would read 1100 1001. It is a very incon-

venient and time-consuming process for a programmer to convert all of the statements into this binary coding and so, in the simplest form of language, the function code is expressed as a mnemonic (an easily remembered group of letters) and the data by its description rather than its address. In this case the above statement would become MUL QTY. Languages that, in the construction of their statements, keep close to the machine code format are known as low level languages.

High level languages are those that can be written as statements in ordinary language (within the rules of the language) that more closely describe the actual problem to be solved – a statement such as 'Add quantity to discount if greater than six'.

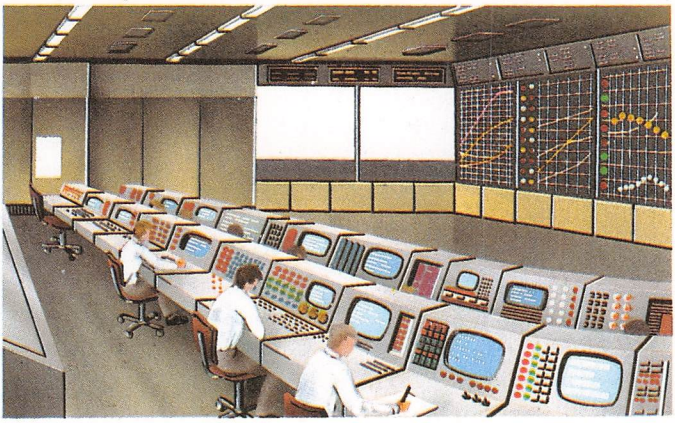
The problem now presents itself as to how these programming statements, written by the programmer in ordinary language (known as a source program), can be converted and stored in the computer in the binary coding (known as an object program) in which it operates. The answer to this is that a translating device is used. This device contains a list of all the permissable statements in a computer language and also the equivalent of each statement in the binary machine code. The whole of the translator is read into the computer store and then, as each source program statement is entered, the equivalent binary code is looked up and output to, say, a magnetic tape. At the end of the process we have an object program, written in machine code, that can, in the future, be entered into the computer when required. The name we give to this translating device in the case of low level languages is an assembler program and for high level languages a compiler program.



Computers in Action

At the beginning of this book it was suggested that it was difficult to get away from the influence of computers in our lives. They have an effect on many things we do and many of the places we go to. One way in which a computer comes very much into its own is in situations where a rapid decision is necessary. Decisions are made in the light of information and calculations relating to a problem, and in some cases it is impossible for a person to obtain the information and make the calculations quickly enough for a decision to be effective. This is the case with every space flight. Imagine a rocket hurtling through space at thousands of miles an hour. All the

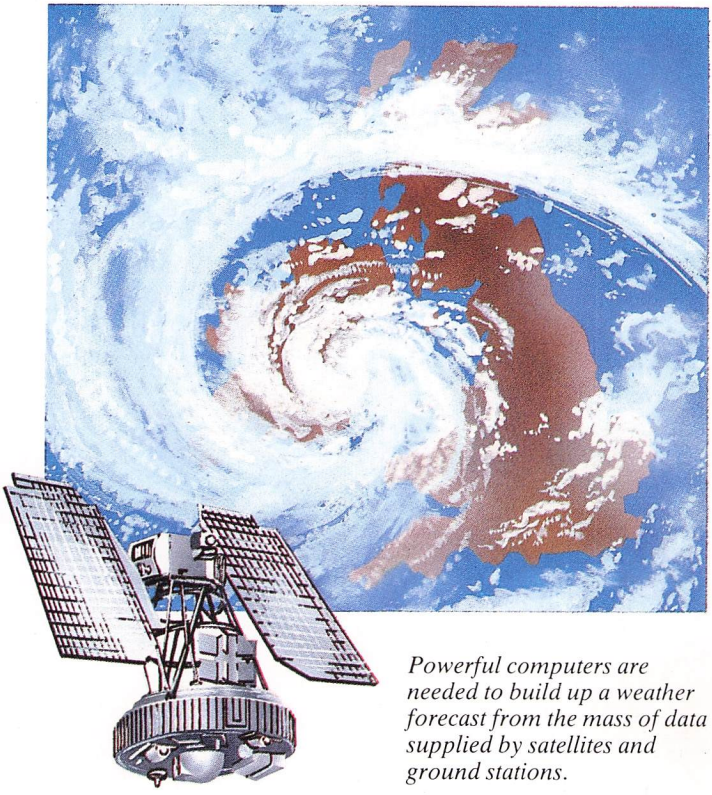
Space flight would not have been possible without the aid of computers. On-board computers monitor all of the spacecraft's systems and flight information, and send a continuous stream of data to computers at mission control.



time its performance is being monitored by instruments inside it and details of speed, direction, fuel, temperature, etc. are being transmitted back to a control centre on the Earth. All these facts are assessed and should it become necessary as a result to modify the rocket's performance in some way, say its direction, then signals are immediately sent back that will automatically set in motion the mechanics to rectify the situation. If it were left to human beings to work out the involved calculations and find the degree of adjustment necessary, it would take so long that by the time a decision was arrived at it would be too late to serve any purpose. The only possible way is to use the incredible speed of a computer. Without computers, space travel would be impossible. The name given to this type of computer processing, where input is received direct from source, the answer worked out, and action instituted in time to have an effect on the occurrence that originally gave rise to the input, is real time working.

Computing the Weather

Another way in which computers play a part in our everyday life is in weather forecasting. The mass of information received from all parts of the globe relating to weather conditions, temperature, winds, barometric pressures, cloud formations, etc., is so vast that it would be impossible without the use of a computer to marshall all of this data, analyse it and draw conclusions in time for the resulting forecast to be of any use. The vast storage capacity of the machine and the speed of the operation enables it to absorb all of this information, sort it all out, compare the resulting statistics with details of similar previous



Powerful computers are needed to build up a weather forecast from the mass of data supplied by satellites and ground stations.

climatic conditions already held in its store, and so arrive at a forecast for the future. It will even use this information to print out weather maps showing the position of isobars, giving temperatures, wind speeds, etc.

Airline Bookings

Apart from these scientific applications, computers are of course also used extensively in business and

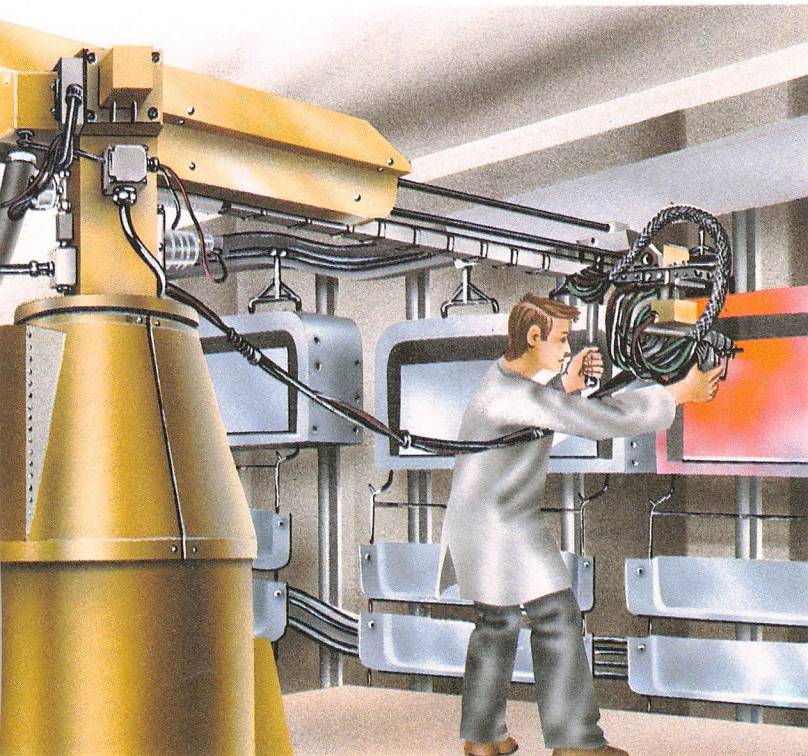
commerce. One of the first major computer systems introduced, and still in use today, is designed for booking airline seats. The airline company keeps details of all of the seats on all of its flights stored in a central computer. To this computer are linked terminals sited all over the world, usually consisting of a keyboard for entering information and a visual display unit on which information can be displayed. When a client wishes to book a seat the operator keys the flight number into the keyboard and almost immediately the available seats on the flight are shown on the visual display. What the computer has done is to look up the details of the flight, probably stored on a magnetic disc (a computer can go directly to any record on disc in about 40 milli-seconds), and transmit these details back to the terminal. When a seat has been booked the computer is informed of this through the keyboard and immediately it brings the records on the disc up to date by noting that the seat is no longer available. This process of keeping records up to date with changing circumstances is called computer up-dating.

Computers have also been extensively developed for industrial purposes. It was soon realised that they could play an invaluable role in operations such as controlling machine tools and monitoring conditions in chemical plants. Today, computer-controlled robots are taking over many of the repetitive tasks of the factory production line. Once the robot has been 'taught' a certain series of operations, its computer will ensure that the work is carried out flawlessly and tirelessly.

Industrial robots can 'learn' how to perform tasks. Once they have been guided manually through an operation (opposite) the information is stored on a memory chip.

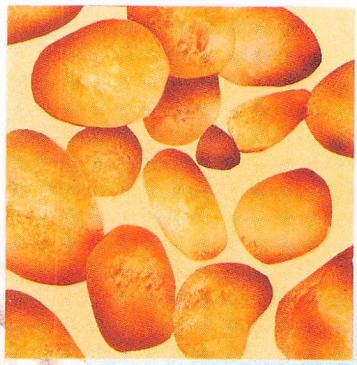
The Magic Chip

The 'magic' of the small device known as a silicon chip lies in the amount of information it can store, the complex calculations it can carry out and the extent to which it is used in controlling machines and events. Like so many other technological developments, it is a by-product of the space age. In the early days of space technology the then existing computers were far too big and heavy to be used for this purpose. What was needed were devices small and light, yet at the same time powerful enough to cope with the demands of complex operations. Thus the silicon chip was born.



Making a Chip

A silicon chip starts life as a very thin wafer of pure silicon about 15 cm in diameter. It is coated with a photo-electric emulsion and then exposed to a light source through a mask divided up by thin lines into about 400 small rectangles. Where the light passes through the rectangular holes the emulsion changes chemically, allowing it to be dissolved away. Each of these small rectangles will eventually become a separate silicon chip. We saw earlier, in the chapter on central processors, how, in principle, a silicon semiconductor works. The pure silicon, which in itself is an electrical non-conductor, is coated, or impregnated, with an element, the choice depending on the type of semi-conductor being produced. For example, it may be impregnated with boron, phos-

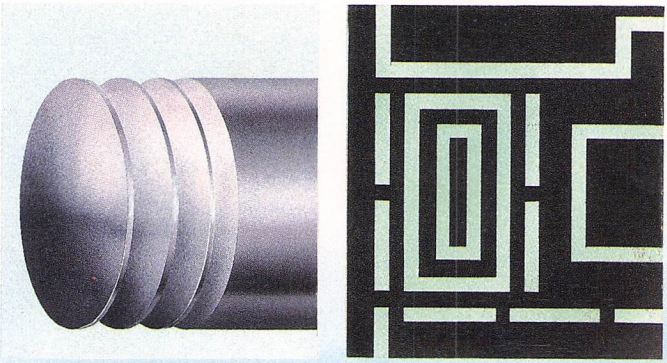


Silicon is one of the most common substances on Earth. Grains of sand (shown enlarged) are mostly silicon.

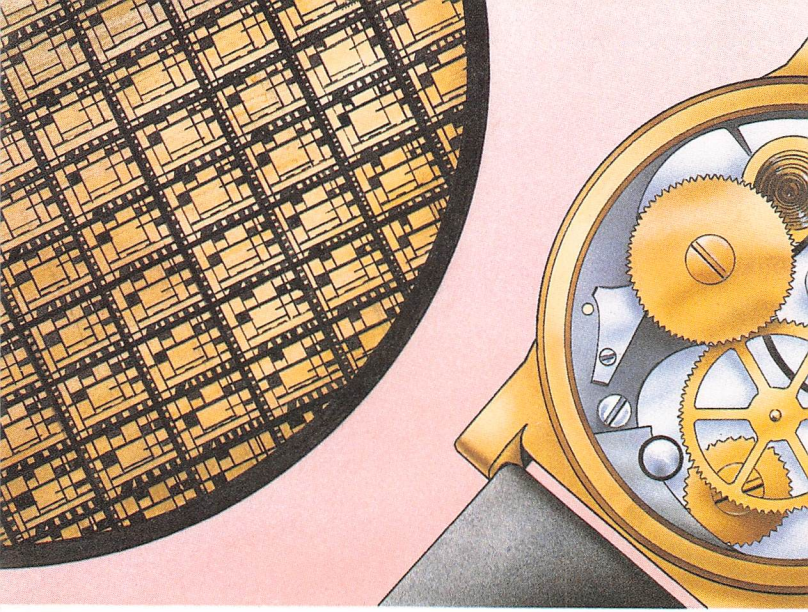
phorus or arsenic in order to give the silicon the property of being able to conduct electricity in one direction only.

Minute connecting wires are condensed on to the chip surface to make a very large number of semi-conductors, say up to 100,000 on a 5 mm square chip. After testing, access lugs (connecting pins) are wired in to the circuitry of the semi-conductors and the whole thing encased in plastic, leaving the connectors protruding from the casing so that the chip can be plugged into the device it is intended to operate.

Nowadays the basic electronics of all calculators and computers is in the form of silicon chips. It is usual to use separate chips with specialized functions for the different operational parts of a computer, although it is possible to build a complete micro-



The first stage in chip processing is to slice perfect crystals into very thin, disc-shaped wafers (above left). The circuitry is designed at a vastly enlarged size on a screen (above right).



A silicon wafer before cutting, showing some of the several hundred chips, compared in size with a watch movement.

computer on one chip. For example, a typical memory chip will contain semi-conductors to hold 16K (i.e. 16 × 1024 = 16,384) binary bits. A microcomputer memory made up of 32k (32,768) 8-bit words would consist of 16 such chips. Another chip will contain the central processor arithmetic and logic circuitry, another the computer's clock (this is a timing device which controls the movement of the binary bits from one section of the processor to another), and yet another may contain the assembler used in programming to convert the source program into an object program.

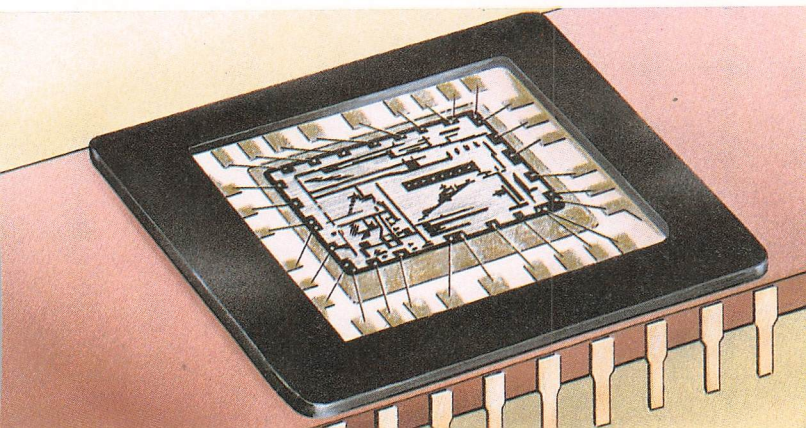
One important thing about semi-conductor memory chips is that they may be of two types. On the one hand the state of the semi-conductors may be

capable of being changed at will, i.e. to represent a 0 or a 1 as required. These are the kind used in a computer's basic memory bank. They are called RAM (read and write memory) chips.

On the other hand, chips may be required to store information that will never change. These are known as ROM chips (read only memory). One example of the use of ROM chips is in the control of industrial machinery, for example, a drilling machine. Once the pattern of the machine's movements have been established and stored in a ROM silicon chip, the chip will then automatically control the repetition of this sequence of movements. Another example, a domestic one this time, is the use of ROM chips to control the operations of a washing machine. The chip stores permanently a number of different washing programs, each one of which can be switched on at will.

Not only have silicon chips made an impact on the design and control of computers, industrial machinery and domestic appliances but our leisure time has also felt their impact. We can now spend hours trying

The finished chip is mounted so that it can be plugged in to the device it is intended to operate.

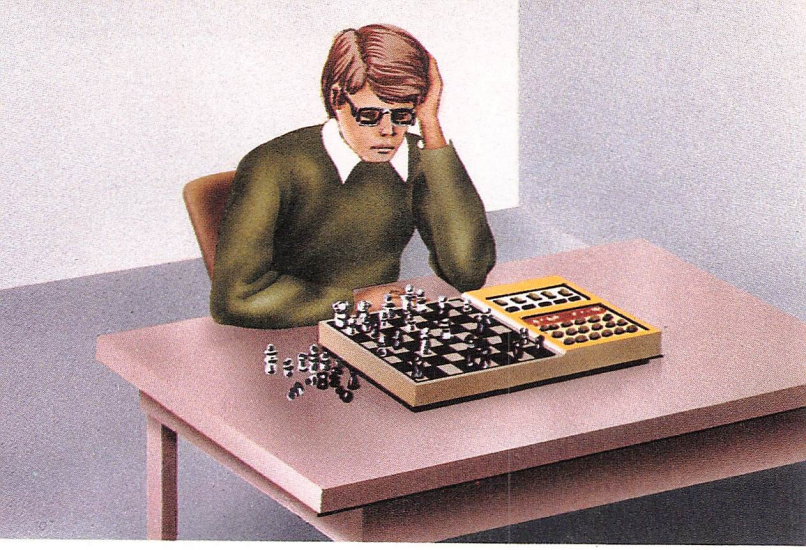


to beat a computerized chess game in which all of the moves to counteract our own are stored in chips. We can sit in front of our television sets playing simulated tennis or football. The chip also plays its part in helping to safeguard our health. Should we be seriously ill we may be looked after in an intensive care unit controlled by microcomputers. Our bodily functions would be continuously monitored through sensors attached to different parts of our body and an alarm would immediately sound should anything go wrong.

The Chip Tomorrow

It is difficult to forecast the future impact of the silicon chip. From a technical point of view we can expect chips to become more and more densely packed with electronic components and therefore more and more powerful. Even today's small desk computer, based on chip technology, can be more powerful than the large main-frame machine of 10 years ago, with its need for a large air-conditioned room to house it and costing probably a hundred

The 'magic' chip has created desk-top computers with the power of a room-size machine 20 years ago.



Microcomputers are increasingly used for games – chess being a good example. But the computer's chess game is constrained by the limitations of its program.

times as much. If the processing power of microcomputers increases over the next 10 years at the same rate as the past 10 we can expect in the 1990s to be able to carry around in our pocket a machine more powerful than the total power of thousands of today's machines. Of this one thing we can be certain: over the next few years the silicon chip is going to materially expand the degree of automation in our lives. Will we, in the foreseeable future, be able to take a computer out of our pocket, talk to it and have it talk back to us? Will we during the next few years be able to dispense with money and have the cost of everything we buy automatically debited to our bank account through a chip-controlled sales recording device? Both are possible. Many people think that the chip is still in its infancy; who knows what it will do when it comes of age.

Glossary of Terms

Accumulator A special register in the central processor in which a number can be temporarily held while arithmetic is being performed.

Adder The electronic circuitry in a computer that carries out arithmetic and logical operations.

Address A unique reference given to each word in the central processor memory or each sector in a direct access backing storage device.

Arithmetic and Logical unit That section of the central processor that organizes, controls and carries out arithmetic and logical processes.

Assembler A special program for converting source language statements of low level languages into equivalent machine code statements.

Backing Storage Storage that is located outside the central processor but connected to it electrically, and to and from which data can be transferred to the processor.

Bar code A pattern of printed lines in binary coding that can be read into a computer by light pen scanning.

Binary A number system having a radix of two and using two digits only – '0' and '1'.

Bit A single binary digit, '0' or '1'.

Cassette Tape A magnetic tape, similar to those used on domestic tape recorders, designed for use with microcomputers.

Central Processing unit This is the main part of a computer in which all processing is carried out and which controls the activities of the whole computer configuration.

Chip A small electronic device containing many thousands of components, having a silicon base.

Compiler A special program for converting high level computer language statements into machine code.

Data All of the facts and figures relating to a situation or a system.

Disc A form of backing storage. A magnetic disc on which data or programs may be stored and referred to as and when required.

Floppy Disc A small magnetic disc designed for use with microcomputer systems.

Flow Chart A chart illustrating the logical sequence of events that must be performed to attain a pre-determined aim.

Gate An electronic switching device.

Hardware A collective term used for all of the machines making up a computer configuration.

Input The process of conveying information to a computer, usually through some mechanical device.

Instruction The basic unit of a computer program. A program may consist of many instructions of varying types.
Jump A technique used in programming to direct computer control from one section of the program to another.
Keyboard A type of input device operated by depressing alpha/numeric keys.
Machine code The pattern of binary bits stored in a computer that indicate and cause the performance, when quoted, of specific machine functions.
Memory Refers to the internal storage capacity of the central processor. Also known as an immediate access store.
Microcomputer A small computer based on microelectronic technology.
MICR Magnetic ink character recognition. An input technique that uses the magnetic qualities of the ink in which the characters are printed.
OCR Optical character recognition. An input technique based on the process of light scanning stylized characters.
Output The process of communicating the results of the computer's processing, often through a printer or a visual display unit.
Peripherals A collective term for all input, output and storage devices external to, but connected with, the central processor.
Printer A device used for computer output. A line printer prints a whole line of type at a time, a serial printer one character at a time.
Program A series of instructions that can be stored in the processor and that will be executed by it in sequence.
Punch Card A card in which information can be stored by punching holes through it.
RAM Read and write memory units, the contents of which can be changed as required.
ROM Read only memory units, the contents of which remain static.
Silicon The material which provides the base for solid state microelectronic components.
Software A collective term referring to all the programs used on a computer.
Storage Areas internal or external to the central processor in which data and programs can be stored.
Terminal A device, often consisting of a keyboard and/or visual display unit that is connected to a computer and can be used for the remote entry and receipt of information.
Visual Display Unit Similar to a television screen on which text or diagrams can be displayed as output from a computer.
Word A sub-division of the central processor store containing a standard number of bits. It is individually addressable.

Index

Abacus 8
Address 33, 34
Airline reservations 53, 54
AND gate 37–40
Arithmetic and logic unit 16, 36
Assembler program 49

Babbage, Sir Charles 9
Backing storage 15, 16, 41–44
Bar code 6, 27
Binary bit 35
Binary coded decimal 23
Binary decimal conversion 22
Binary number system 19–23

Central processor 15, 16, 30–40
Compiler program 49
Computer, arithmetic 36–40
  configuration 13
  development of 8–12
  elements of 13–17
  language 48–50
  programs 45–49
  up-dating 54

Daisy wheel printer 28, 29
Data processing 14
Data storage 41–44
Decimal binary conversion 22
Direct access storage 42–44

Electronic pen 25
ENIAC 10, 11
Exclusive OR gate 38–40

Ferrite cores 30–33
Floppy discs 44
Full-adder 39, 40
Function code 40

Gates 36–40

Half-adder 38–40
Hardware 13
High-level language 49

Immediate access store 16
Impact printing 28, 29
Input devices 15, 16, 24–27

Keyboard devices 27, 28
Keypad 27

Line printer 28
Logical operations 36–40
Logic flow charts 45–47
Low-level language 49

Machine instruction format 48
Magnetic discs 42–44
Magnetic ink character recognition 26, 27
Magnetic tape 41, 42
Main frame computer 16, 17
Markers 42
Memory chip 58
Microcomputer 12, 16
Micro-electronics 12
Microfiche 29
Microfilm 29
Minicomputer 16, 17

Napier, John 8
Non impact printing 28, 29
NOT gate 37–40

Object program 49
Operation code 40
Output devices 15, 16, 28, 29

Pascal, Blaise 8, 9
Peripherals 16
Photographic output 29
Program instruction 40
Punched cards 24–26
Punched tape 24–26

RAM chips 59
Random access storage 42–44
Reading devices 24–27
Real time working 52
ROM chips 59

Semi-conductors 31, 35
Serial access storage 42
Serial printer 28
Silicon chip 12, 55–61
Software 13
Software program 49
Space flight 50–52
Speech recognition 25

Transistors, invention of 11, 12
Translating device 49

Visual display unit 28, 29

Weather forecasting 52, 53
Word 33, 34

Granada Guides are an invaluable series of compact reference books, perfect for bookshelf or pocket. Each handy volume contains the latest expert information on its subject, together with explanatory diagrams, brilliant colour illustrations, and a comprehensive index.

TITLES INCLUDE

FLAGS
HORSES AND PONIES
WARPLANES
DINOSAURS
SPACECRAFT
STARS AND PLANETS
TANKS
MOTORCYCLES
DOGS
HUMAN BODY
ROCKETS AND MISSILES
RACING CARS
TREES
WILD FLOWERS
WILD ANIMALS
GARDEN BIRDS
ANCIENT GREECE
ANCIENT ROME
ANCIENT EGYPT
ANCIENT CHINA
AIRLINERS
ROBOTS
COMPUTERS
FOSSILS

IN PREPARATION

SEASHORE
INSECTS
FISHES
MUSHROOMS AND TOADSTOOLS

GRANADA PUBLISHING
0 246 11895 4

£1.95 net